SDM COLLEGE OF ENGINEERING AND TECHNOLOGY.

DHAVALAGIRI, DHARWAD. 580002

Project Title

Optical Character Recognition Using Neural Networks.

Team

Madhusoodan M Pataki	2SD14CS054
Anarghya K Jantali	2SD14CS128

Guide

Prof .Sharada H N

Abstract

Optical Character Recognition (OCR) deals with recognition and conversion of printed characters in a scanned document that is taken as an input to an editable, digitized text. This improves the storage, processing, mining that are pivotal in further development of essential applications that serve various domains. We propose a process, which takes a scanned/captured image as an input, pre-process the image, recognize the characters of English language and output in a form of string to a text document. Current scope is limited to English characters and digits. Recognition is also limited to printed characters rather than manuscript/handwriting. This process can be improvised to recognize various languages as well as handwritten materials. The input image is put through pre-processing, segmentation and feature extraction. Feature extraction and recognition of characters is implemented with Neural-Networks. Using neural network to train and recognize the characters increases the coverage as well as accuracy of the results compared to conventional horizontal and vertical feature extraction methods. With increased accuracy, this system can be implemented in applications such as form reading such as cheque, postal code, official letters etc. This can also be augmented with a dictionary and intelligent interpreter to evaluate answer scripts etc.

Table of contents:

1. Introduction.

Character recognition is one of the problem in computing. Due to sequential nature of the computing, this has been a conventional hurdle in image processing and pattern recognition domain. Several works have been published in regards to improve processing time and recognition accuracy. Drastic improvements have been made in two prominent methods that are pattern recognition of characters as well as feature extraction. Pattern recognition in general, has an inherent limitation to the model due to decreased accuracy, when faced with new font/handwriting. Maturing science of neural networks has made it possible to input a feature to neural- network, proving classification of characters with better accuracy.

Character recognition proposed in this paper is offline-recognition. An image is given to the system after it is scanned/captured. This also, involves problems such as noise, orientation, bleaching etc. As such, pre-processing of the image is a necessity in offline recognition systems. Pre-processing involves steps that are required to shape image into suitable form for segmentation. This step involves skewing, noise-reduction, bleaching etc. The cleaned up image is then given as input to segmentation

Segmentation is a process of identifying individual glyphs present in the image. The characters are read line by line, spaces, punctuations and characters are identified separately. The character glyphs are normalized to a standard pixel size, so that extraction of features of character glyphs is made easy.

The Selection of appropriate feature extraction method is probably the single most important factor in achieving high recognition performance. Features can be classifies into two sets, statistical features and structural feature. In statistical feature, an image is decomposed into a set of n dimensional identifiers. In structural featuring, end points, intersections, convexity, strokes etc. define the geometrical attributes of a character. Several methods of feature extraction for character recognition have been reported in the literature [2]. The widely used feature extraction methods are Template matching, Unitary Image transforms, Projection Histograms, Contour profiles, Zoning, Nearest-Neighbor rounding etc. The features extracted are mapped to characters to train the neural network.

An artificial neural Network as the backend is used for performing classification and recognition tasks. Classification techniques have been applied to handwritten character recognition since the 1990s. These methods include statistical methods based on Bayes decision rule, Artificial Neural Networks (ANNs), Kernel Methods including Support Vector Machines (SVM) and multiple classifier combination [3], [4].

1.1 Literature Survey

1. Dinesh Dileep et al, "A feature extraction technique based on character geometry for character recognition"
   This paper describes a geometry based technique for feature extraction for segmentation based recognizing systems.

2. J Pradeep et al, "Diagonal based feature extraction for handwritten characters"
   This paper proposes a new feature extraction method that normalizes the characters into diagonal zones consisting of uniform pixel density.

3. M. Zahid Hossain M. et al. “Rapid Feature Extraction for Optical Character Recognition”
   This paper describes the different feature extraction methods and algorithms.

4. Simon Haykin “Neural Networks - A Comprehensive Foundation”
   This book has immense explanation on Neural Networks.

2. Methodology

This section is further divided into the following sections.

1. Implementation details.
2. Working of OCR & NN.
3. Limitations

2.1 Implementation details.

2.1.1 Language chosen and external libraries.

Java was chosen as the language for implementation. The reason is simple, it’s cross platform, Object Oriented and have lot of utilities for basic image processing such as for reading different image formats, simple and cross–platform utilities for GUIs which help us visualizing things and also “we know it better”. No external libraries other than javalibs will be used for the development as Java has large collection of libraries and packages built for processing images.

2.1.2 Tools used.

Some tools for visualizing the segmentation, binarization and feature extraction were built by us while some are from internet which help in understanding Neural Networks.

2.2. Working of OCR and NN.

The working of OCR can be divided into the following phases which are explained in detail below.

1. Learning phase.
2. Working/Testing phase.

2.2.1 Learning phase

The Neural Network is trained for recognizing the characters in this phase. This phase further contains the phases shown in the image below and are explained in following sections.

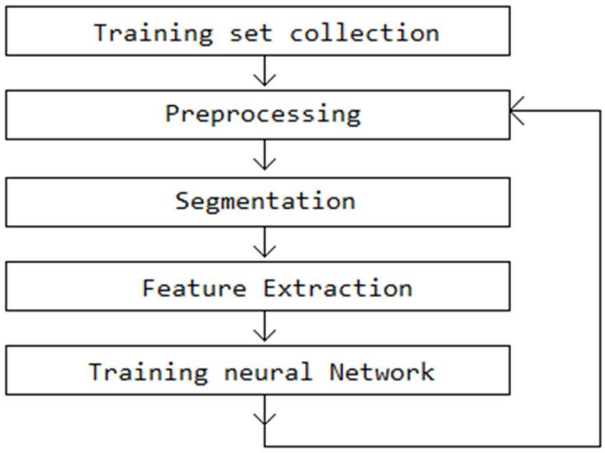

Figure 2.1 Block diagram of learning phase of NN.

2.2.1.1 Training set collection: Here the different types of input images are collected and are labeled with the expected output. We store these images in folders named by label. The reason for many images is as the number of images increase more will the variation in input and learning will be efficient and recognition accuracy increases. The criteria for selection of training set include

a. Noisy images
b. Skewed Images.
c. Simple but different fonts
d. Different sized fonts.

2.2.1.2 Preprocessing

This step involves the basic transformations on the input image such as

a. Noise removal: The image may have some noise associated with it if it has been captured by some device as camera. The noise in the image can reduce the accuracy hence it's required to remove the noise.
b. Skew detection and correction: The image may be skewed to some degree which may also affect the accuracy of the system hence we do the skew detection and correction.
c. Binarization: The foreground plane is represented by 0 and background plane by -1 to minimize the complexity of processing the image. This is done by checking the amount of colors at corners and checking for the character sized color blocks in the image. Since these are not the primary focus of us we will do Noise removal and Skew Detection and Correction by some APIs available.

2.2.1.3 Segmentation

The image is segmented in lines, words and characters. The line segmentation is done by analyzing the histogram of vertical gaps between foreground colors while the word and character segmentation is done by analyzing histograms of horizontal gaps between fore-ground colored components. The segmented characters are now input to the feature extraction.

2.2.1.4 Feature Extraction

The characters can be visualized by features. There are many features known to be used such as

1. Crossing: Number of transitions from foreground to background and vice-versa along a straight line.
2. Projection Histograms: The number of foreground pixels in scanned lines are used as features here
3. Pixel density of zones : number of foreground pixels per zone
4. Structural components: Number of different types of lines, curves and connectivity between them
5. Polynomials: Constructing the polynomial which best fits the curve is also an option.

We are not yet sure which features to choose as there are many and we are stuck in comparing them. The extracted features are sent to the Neural Network for classification and learning.

2.2.1.5 Learning by Classification.

The Neural network designed is a multilayer feedforward classifier with multi-dimensions. The dimensions here represent a class of feature such as connectivity between structural components. The NN classifies the input by its features (dimensionality reduction of feature vector) and stores them along with the labels associated with it. The classification is done at several levels with based on some criteria and a threshold for passing that criteria. The feature vectors are stored in the files based on the features for future matching. Below is an example of the classification levels.

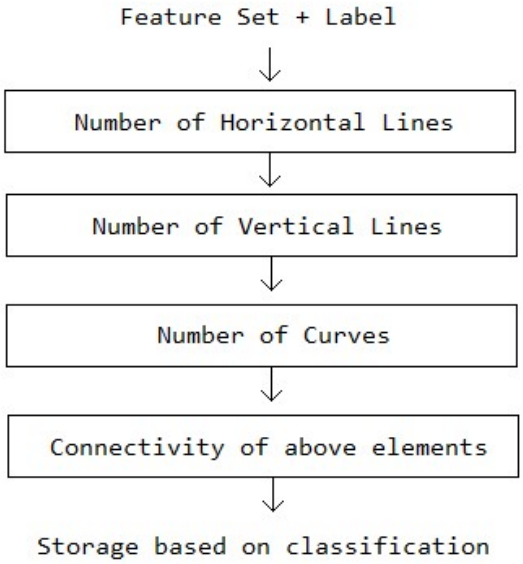

Figure 2.2 Example for learning

2.2.2 Working/Testing phase.

In this phase the Neural Network is used to recognize the characters and is verified for correctness. This helps in making the classification better and increasing the accuracy of recognition.

This also involves some of the above steps in learning phase but with features with no labels. The block diagram of the working phase is as given below. These steps are explained in further sections.

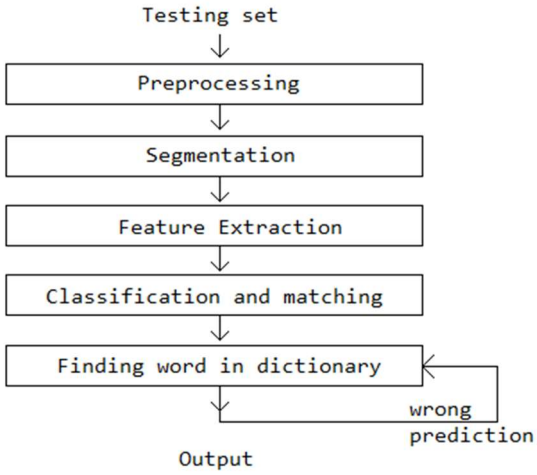

Figure 2.3 Block diagram of test phase.

2.2.2.1 Testing set

Here the testing set includes just images with no labels.

2.2.2.2 Preprocessing

Similar to the ones in learning phase.

2.2.2.3 Segmentation

Similar to the ones in learning phase.

2.2.2.4 Feature Extraction

Similar to the ones in learning phase.

2.2.2.5 Classification and Matching

The features input are classified similarly as done in learning phase. At each level some of the features match and only the ones who cross the threshold are selected for further matching. These are input to next level with the probability of match with a label. Below is an example.

Consider the classifier in the Fig 3.2 and the character image input is of the character ‘d’.

The feature set will be

1. A right opening curve connected to a vertical line at two points.
2. A vertical line connected to a curve at two points.

In first three levels say the characters ‘a’(in handwritten format), ‘d’ and ‘q’ match with ‘d’ as they all have one vertical line and a right opening curve. In the next level the connections with distance will be checked and say ‘a’ will be ruled out.

After these levels say the characters ‘q’ and ‘d’ are left which are used to construct the words which are further checked in the dictionary for last verification. The one with dictionary match and high probability will be selected.

2.3 Limitations

These are the limitations of the implementation

1. Limited to some simple fonts.
2. Accuracy is not so good for noisy and skewed images.
3. Non reusable NN as built for specific application.

3. Outcomes

3.1 Student Specific Outcomes.

1. We Learn techniques for
    a. Processing Images
    b. Different types of feature extraction methods.
2. Appreciate the working of human brain in processing real time data in superfast manner.
3. Learn to try how to mimic human brains complexity in computers by implementing Neural Networks

3.2 Project Specific Outcomes.

1. A standalone trained Neural Network library which can be used for development of smart applications such as.
    a. Document Digitizer.
    b. A book reader for blind people when some books are not converted to brail script.
    c. Number plate scanner for the Tolls.
2. A codebase to learn Neural Networks better.

4. References

[1]. "Pattern Recognition and Machine Learning" by Cristopher M. Bishop. ISBN- 13:978-0387-31073-2 Springer Science and Business Media LLC.

[2]. Seethalakshmi R et al. "Optical Character Recognition for printed Tamil text using Unicode". Journal of Zhejiang University SCIENCE ISSN 1009-3095

[3]. Usha Tiwari et al. "Text Extraction from Images". International Journal of Electronic and Electrical Engineering. ISSN 0974-2174Volume 7, Number 9 (2014), pp. 979-985

[4]. Sandhya Arora et al. "Combining Multiple Feature Extraction Techniques for Handwritten Devanagari Character Recognition" 2008 IEEE Region 10 Colloquium and the Third ICIIS, Kharagpur, India

[5]. Gaurav Y. Tawde, and Jayashree M. K. International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622

[6]. Rafael C. Gonzalez, Richard E. woods and Steven L. Eddins. Digital Image Processing, Pearson Education, Dorling Kindersley, South Asia, 2004.

[7]. Anil. K. Jain and Torfinn Taxt, “Feature extraction methods for character recognition-A Survey” Pattern Recognition, vol. 29, no. 4, pp. 641-662, 1996

[8]. C. L. Liu, H. Fujisawa, “Classification and Learning for Character Recognition: Comparison of Methods and Remaining Problems”, Int. Workshop on Neural Networks and Learning in Document Analysis and Recognition, Seoul, 2005.

[9]. J Pradeep et al, "Diagonal based feature extraction for handwritten characters", International Journal of Computer Science & Information Technology (IJCSIT), Vol 3, No 1, Feb 2011

[10]. Dinesh Dileep et al, "A feature extraction technique based on character geometry for character recognition", International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 2, 2014.

[11]. M. Zahid Hossain et al. “Rapid Feature Extraction for Optical Character Recognition”